

Amendments to the Claims

1. (currently amended) A networked system comprising:

a network;

a server coupled to the network, wherein the server includes:

an application code source that stores a client application; and

a server code manager coupled to the application code source; and

a client coupled to the network said client not having said client application stored thereon, wherein the client comprises:

a CPU for natively executing code segments derived from the client application stored on said server;

a code cache coupled to the CPU, for storing code segments; and

a client code manager coupled to the code cache, wherein the client code manager launches the client application by requesting that the server code manager transmit a code segment to the client, wherein a granularity of the code segments is dynamically tailored to the client to balance server-side and client-side execution and client-side storage requirements, and is configured based on predicted code segment usage or prior code segment usage history, receiving the code segment from the server, storing the code segment in the code cache, and executing the code segment using the CPU until the code segment attempts to pass control to a required code segment not stored in the code cache, at which point control passes back to the client code manager to retrieve the required code segment from the server, with the CPU continuing execution with the required code segment.

2. (original) The networked system of claim 1 wherein the server code manager comprises:

an application code transformation manager coupled to the application code source, for transforming the client application from a first format to a native

binary format compatible with a native instruction set of the CPU of the client;
and

a server code segment manager coupled to the application code transformation manager, for parsing the client application in the native binary format into code segments that are transmitted to the client via the network.

3. (original) The networked system of claim 2 wherein the first format is a native binary format other than the native binary format of the CPU of the client, and the application code transformation manager comprises a transformation engine to transform the client application from the first format to the native binary format of the CPU of the client.

4. (original) The networked system of claim 2 wherein the first format is a source code text format of a programming language, and the application code transformation manager comprises a compiler that compiles and links the client application into a native binary format of the CPU of the client.

5. (original) The networked system of claim 2 wherein the first format is a virtual machine format, and the application code transformation manager comprises a just-in-time compiler that compiles and links the client application into a native binary format of the CPU of the client.

6. (original) The networked system of claim 1 wherein the client code manager comprises:

a client code segment manager coupled to the network and the code cache, wherein the client code manager requests needed segments from the server; and

a code cache linker and manager coupled to the code cache and client code cache segment manager, wherein the code cache linker and manager links the code segment received from the server into the code cache, emits the received code segment into the code cache, and branches to the received code

10003355-1

-3-

Serial No.: 09/874,170

Examiner: Proctor, J.

Art Unit: 2123

segment the code cache.

7. (previously presented) The networked system of claim 6 wherein the code cache linker and manager further comprise:

adjusting any branch targets in code segments stored in the code cache that need to branch to the received code segment and had previously been adjusted to branch out of the code cache to the client code segment manager to now branch to appropriate locations within the received code segment;

adjusting any branch instructions in the received code segment having branch targets that branch to code segments currently in the code cache to branch to the appropriate code segments in the code cache; and

adjusting any branch instructions in the received code segment having branch targets that need to branch to code segments not in the code cache to branch out of the code cache to the client code segment manager to request the code segment containing the branch targets.

8. (currently amended) The networked system of claim 6 wherein the code cache linker and manager includes a code cache maintenance unit coupled to the code cache, for removing old and unneeded code segments from the code cache, replacing older segments with newly received segments when the code cache reaches a certain threshold, ~~and optimizing code segments in the code cache.~~

9. (currently amended) A server comprising:

a network interface that couples the server to a network, which in turn is coupled to a client;

an application code source that stores a client application; and

a server code manager coupled to the application code source and the network interface, wherein the server code manager derives native binary code segments in a native execution format required by client CPUs from the application code source, wherein a granularity of the code segments is dynamically tailored to the client to balance server-side and client-side execution

10003355-1

-4-

Serial No.: 09/874,170

Examiner: Proctor, J.

Art Unit: 2123

and client-side storage requirements, and is configured based on predicted code segment usage or prior code segment usage history, and transmits the native binary code segments to said client not having said client application stored thereon upon requests received from said client.

10. (previously presented) The server of claim 9 wherein the server code manager comprises:

an application code transformation manager coupled to the application code source, for transforming the client application from a first format to the native execution format; and a server code segment manager coupled to the application code transformation manager, for parsing the client application in the native execution format into code segments that are transmitted to said client via the network.

11. (original) The server of claim 10 wherein the first format is a native execution format other than the native execution format required by a client CPU, and the application code transformation manager comprises a transformation engine to transform the client application from the first format to the native execution format required by the client CPU.

12. (original) The server of claim 10 wherein the first format is a source code text format of a programming language, and the application code transformation manager comprises a compiler that compiles and links the client application into a native binary format required by a client CPU.

13. (original) The server of claim 10 wherein the first format is a virtual machine format, and the 2 application code transformation manager comprises a just-in-time compiler that compiles and links the client application into a native binary format required by a client CPU.

14. (currently amended) A client comprising:

10003355-1

-5-

Serial No.: 09/874,170

Examiner: Proctor, J.

Art Unit: 2123

a network interface that couples the client to a network, which in turn is coupled to a server;

a CPU for natively executing code segments derived from the client application stored on the server, wherein a granularity of the code segments is dynamically tailored to the client to balance server-side and client-side execution and client-side storage requirements, and is configured based on predicted code segment usage or prior code segment usage history;

a code cache coupled to the CPU, for storing code segments; and

a client code manager coupled to the code cache, wherein the client code manager launches the client application stored on said server and not on said client by requesting that the server transmit a code segment to the client, receiving the code segment from the server, storing the code segment in the code cache, and executing the code segment using the CPU until the code segment attempts to pass control to a required code segment not stored in the code cache, at which point control passes back to the client code manager to retrieve the required code segment from the server, with the CPU continuing execution with the required code segment.

15. (original) The client of claim 14 wherein the client code manager comprises:

a client code segment manager coupled to the network and the code cache, wherein the client code manager requests needed segments from the server; and

a code cache linker and manager coupled to the code cache and client code cache segment manager, wherein the code cache linker and manager links the code segment received from the server into the code cache, emits the received code segment into the code cache, and branches to the received code segment in the code cache.

16. (previously presented) The client of claim 15 wherein the code cache linker and manager further comprise:

adjusting any branch targets in code segments stored in the code cache that need to branch to the received code segment and had previously been adjusted to branch out of the code cache to the client code segment manager to now branch to appropriate locations within the received code segment;

adjusting any branch instructions in the received code segment having branch targets that branch to code segments currently in the code cache to branch to the appropriate code segments in the code cache; and

adjusting any branch instructions in the received code segment having branch targets that need to branch to code segments not in the code cache to branch out of the code cache to the client code segment manager to request the code segment containing the branch targets.

17. (currently amended) The client of claim 15 wherein the code cache linker and manager includes a code cache maintenance unit coupled to the code cache, for removing old and unneeded code segments from the code cache, replacing older segments with newly received segments when the code cache reaches a certain threshold, ~~and optimizing code segments in the code cache.~~

18. (currently amended) A method of executing an application in networked system comprising:

at a client: issuing a code segment request to a server coupled to the client by a network;

at the server: receiving the code segment request from the client;

deriving a code segment in a native execution format required by the client from an application code source stored on said server and not on said client, wherein a granularity of the code segments is dynamically tailored to the client to balance server-side and client-side execution and client-side storage requirements, and is configured based on predicted code segment usage or prior code segment usage history; and

transmitting the code segment to the client; and

at the client: receiving the code segment;

adjusting branches in the code segment having targets not in a code cache of the client to cause code segments containing the targets to be requested from the server;

emitting the code segment into the code cache; and

executing the code segment natively from the code cache.

19. (original) The method of claim 18 wherein deriving a code segment in a native execution format required by the client from an application code source comprises transforming the application source from a native execution format other than that required by the client into the native execution format required by the client.

20. (original) The method of claim 18 wherein deriving a code segment in a native execution format required by the client from an application code source comprises compiling and linking the application code source from a code text format of a programming language into the native execution format required by the client.

21. (original) The method of claim 18 wherein deriving a code segment in a native execution format required by the client from an application code source comprises using a just-in-time compiler to compile the application code source from a virtual machine format into the native execution format required by the client.

22. (original) The method of claim 18 wherein the code segment is a first code segment, and executing the code segment natively from the code cache includes:

at the client: executing a branch in the first code segment that seeks to branch to a second code segment not in the code cache; and

issuing a code segment request for the second code segment to the server;

at the server: receiving the code segment for the second code segment from the client; deriving the second code segment in the native execution format required by the client from the application code source; and

transmitting the second code segment to the client; and

at the client: receiving the second code segment;

adjusting any branches in the first code segment stored in the code cache that need to branch to the second code segment and had previously been adjusted to cause the second code segment to be requested from the server to now branch to appropriate locations within the second code segment; adjusting any branches in the second code segment having targets in the first code segment in the code cache to branch to the appropriate location within the first code segment; adjusting any branches in the second code segment having branch targets in code segments not in the code cache to cause the code segments not in the code cache to be requested by the server; emitting the second code segment into a code cache; and

continuing execution in the second code segment at the location to which the first code segment attempted to branch.

23. (currently amended) A computer program product, comprising:

at least one computer usable medium having computer readable code embodied therein for causing an application to be executed in a networked system, the computer program product including:

first computer readable program code devices configured to cause a client to issue a code segment request to a server coupled to the client by a network;

second computer readable program code devices configured to cause the server to receive the code segment request from the client, derive code segment in a native execution format required by the client from an application code source stored on said server and not on said client, wherein a granularity of the code segment is dynamically tailored to the client to balance server-side and client-side execution and client-side storage requirements, and is configured

based on predicted code segment usage or prior code segment usage history,
and transmit the code segment to the client; and

third computer readable program code devices configured to cause the client to receive the code segment, adjust branches in the code segment having targets not in a code cache of the client to cause code segments containing the targets to be requested from the server, emit the code segment into the code cache, and execute the code segment natively from the code cache.

24. (original) The computer program product of claim 23 wherein the code segment is a first code segment, and the third computer readable program code devices includes:

fourth computer readable program code devices configured to cause the client to execute a branch in the first code segment that seeks to branch to a second code segment not in the code cache, and issue a code segment request for the second code segment to the server;

fifth computer readable program code devices configured to cause the server to receive the code segment for the second code segment from the client, derive the second code segment in the native execution format required by the client from the application code source, and transmit the second code segment to the client; and

sixth computer readable program code devices configured to cause the client to receive the second code segment, adjusting any branches in the first code segment stored in the code cache that need to branch to the second code segment and had previously been adjusted to cause the second code segment to be requested from the server to now branch to appropriate locations within the second code segment, adjust any branches in the second code segment having targets in the first code segment in the code cache to branch to the appropriate location within the first code segment, adjust any branches in the second code segment having branch targets in code segments not in the code cache to cause the code segments not in the code cache to be requested by the server, emit the second code segment into a code cache, and continue execution in the second

code segment at the location to which the first code segment attempted to branch.